이력 영상의 시간 간격과 연관성에 의한 데이터 관리 기법

윤 홍 원[†]

요 약

본 논문에서는 기존 의료 영상 이동 방법에서 생기는 문제점을 해결하기 위해서 의료 영상 데이터의 관리 기법을 제안하였다. 의료 영상 데이터의 관리 기법으로써 EAT(Expanded Average Transaction time) 데이터 이동 기법과 시간 연관성 기반 데이터 저장 방법을 제안하였다. EAT 데이터 이동 기법에서는 각 저장 영역에 저장되는 개체 버전을 구분하는 경계값과 각 영역에 저장되는 개체 버전을 정의하였다. 시간 연관성에 기반한 데이터 저장 방법에서는 임의 두 개체 버전에 대한 겹침의 정도와 간격의 정도를 정의하였고, 두 값을 통합하여 개체 버전을 저장 장치에 배치하는 방법을 보였다. 시간 질의의 비율을 변화시키면서 클러스터의 참조 회수를 비교한 실험에서 기존의 방법보다 제안한 방법의 클러스터 참조 회수가 적게 나타났다.

Management of Historical Images by Time Interval and Interrelation

Yun, Hongwon[†]

ABSTRACT

In this paper, we proposed management strategy of medical image data in order to solve the problem in traditional medical images migration method. As management strategy of medical image data we proposed EAT(Expanded Average Transaction time) data migration method and data storing method based on temporal interrelation. In EAT data migration strategy, we define the dividing criterion which distinguish entity versions to be stored in each storage and also define entity versions to be stored in each storage. We defined degree of overlap and degree of difference for any two entity versions, and integrated those values and described method which place entity versions to storage. In order to compare the number of cluster references when we change rate of temporal queries, the number of cluster references of proposed method is smaller than that of traditional method.

1. 서 론

PACS(Picture Archiving and Communication System)는 영상 획득장치, 영상 전송망, 영상 저장 장치 및 데이터베이스, 그리 영상 표시장치로 나누어 지며 각각의 하드웨어와 소프트웨어로 구성된다 [1-3]. 영상 획득장치는 영상 진단장치, 단층촬영장치 등의 다양한 영상 장비로부터 영상을 획득하는 부분이고, 영상 전송망은 영상이라는 용량이 큰 데이터를 전송하기 위한 네트워크 환경을 말한다. 영상 표시장치는 X-Ray와 같은 고화질의 영상을 나타내는 장치이고, 영상 저장 장치는 의료 영상의 효율적인 검색과 저장을 위하여 단기 저장 장치와 장기 저장 장치로 나누어 영상을 저장한다. 단기 저장 장치는 병원에 따라 다르나 2 주일 정도의 영상을 자기 디스크에 온라인으로 저장하고, 단기 저장 장치에서 일정한 보존 기간이 경과한 영상이나 단기 저장 장치의 용량이 부족한 경우에 자기 디스크에서 삭제하고 장기 저장 장치로 옮겨서 저장한다[2,3].

600병상 정도를 운영하는 병원에서는 1주일 동안에 100GB이상의 무손실 압축 영상이 생성되는데, 단기 저장 장치에서 영상의 보존 기간이 2주일인 병원에서는 2주일마다 200GB이상의 영상을 장기 저장 장치로 옮기게 된다. PACS를 운영하는 병원에서는

이 연구는 2000년도 신라대학교 연구비로 이루어졌음
[†] 정회원, 신라대학교 컴퓨터정보공학부 조교수

일반적으로 단기 저장 장치는 디스크 어레이를 사용하고 장기 저장 장치는 제 3의 장치로써 DLT나 CD 쥬크 박스를 많이 사용하고 있다[4-6].

어떤 영상이 일정한 기간이 경과하여 단기 저장 장치에서 장기 저장 장치로 옮겨진 뒤에, 환자의 이력 정보를 검색하기 위하여 다시 장기 저장 장치에서 단기 저장 장치로 옮겨지는 일이 자주 발생하므로 저장 장치를 효율적으로 관리하는 계층적 저장 관리 시스템이 필요하다[7-9]. 2주일 단위로 영상을 이동하는 방법처럼, 일정한 보존 기간이 경과되면 단기 저장 장치에서 장기 저장 장치로 영상을 옮기는 지금의 이동 방법에 의하면, 어떤 환자의 의료 영상이 단기 저장 장치에서 장기 저장 장치로 옮겨진 뒤에 비교 진단이나 이력 정보를 알기 위해서 곧 단기 저장 장치인 자기 디스크로 다시 가져와야 할 가능성이 높다.

일정한 보존 기간이 경과하면 자기 디스크에서 제 3의 저장 장치로 영상을 옮기는 획일적인 기존의 방법은 다음과 같은 문제점이 있다.

첫째, 이전에 촬영한 영상이 최근 촬영한 영상과 비교 진단에 쓰이는 경우가 많은데, 이처럼 곧 검색할 가능성이 높은 이전의 영상마저 자기 디스크에서 제3의 장치로 옮겨 놓을 가능성이 크다.따라서 제 3의 저장 장치에서 다시 자기 디스크로 옮겨야 하므로 검색 속도를 저하시킨다.

둘째, 개체 버전들 사이의 시간 연관성과 무관하게 의료 영상이 서로 다른 여러 개의 CD에 흩어져서 저장될 가능성이 크다.서로 관련 있는 이력 영상을 찾기 위해서는 제3의 저장 장치의 특성 때문에 수십 초의 시간이 소요될 수 있다.

지금까지 의료 영상 데이터베이스에서는 일정한 보존 기간이 경과하면 영상을 옮기도록 하고 있으며 병원의 영상에 대한 효율적인 관리 기법에 대한 연구가 거의 없었다. 본 논문에서는 앞에서 기술한 일정한 보존 기간이 경과하면 획일적으로 이동하는 방법에서 생기는 두 가지 문제점을 해결하기 위해서 의료 영상 데이터의 관리 기법을 제안한다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 의료 영상 데이터베이스에서 영상을 관리하는 기존 방법과 문제점을 살펴보고, 3장에서는 본 논문에서 제안하는의료 영상 관리 기법을 살펴본다.4장에서는 본 논문에서 제안하는 의료 영상 관리 기법과 기존 방법에 대해서 성능을 비교 분석하고, 마지막으로 5장에서 결론을 맺는다.

2. 기존의 의료 영상 관리 방법

의료 영상 데이터베이스 시스템의 데이터는 문자, 영상, 그리고 음성 등으로 구성되고, 환자 정보는 환자 번호, 주소, 담당 의사 등의 데이터를 포함한다.환자의 기록에는 환자 이력 데이터와 의료 영상 데이터가 들어가는데 환자의 이력과 관련된 데이터는 주키, 시간 데이터, 키가 아닌 주 데이터, 이차 데이터로 나눌 수 있다. 주 키는 레코드를 구분하고 시간 데이터는 유효 시간, 트랜잭션 시간, 또는 사건 시간 가운데 어떤 것이 될 수 있다. 키가 아닌 주 데이터는 환자의 기본 정보를 나타내고 이차 데이터는 환자의 현재 상태나 시간이 경과하면서 달라진 상태 데이터를 말한다. 이차 데이터는 시간이 지나면서 계속 추가될 수 있는 특성을 가지고 있다.

기존의 PACS에서 한 장의 CR(Computed Radiography)은 무손실압축 시 8MB의 저장 공간을 필요로 한다[10].원격지에 방사선 전문의의 진단 결과를 보내기 위해서 어떤 PACS에서는 음성 데이터가 저장되기도 한다. 촬영을 통해서 진단이 필요한 환자는 적게는 한번의 영상이 저장될 수 있고, 시간이 경과하면서 환자의 변화된 상태를 보기 위해서는 여러 장의 영상 데이터를 가질 수 있다. 방사선과의 영상 판독소견은 의무기록에 관련된 의료법 18조 6항에 따라 5년간 보관하여야 한다. 소아 환자의 영상은 환자의 나이가 21세가 될 때까지 보관하고, 유방 촬영은 평생 관리한다[11]. 의료 영상 데이터의 보관 기간이 길고 데이터 양이 방대하므로 계층적인 저장 구조로 저장 관리하는 것이 바람직하다.

그림1은 의료 영상 데이터의 계층 저장 구조를 나타낸 것이고, 단계별로 데이터 보존 기간을 표시하였다. 그림1에 표시한 단계별 보존 기간은 병원마다 꼭 같은 것은 아니고 약간의 차이가 있다. 계층 저장 구조에서는 일정한 보존 기간이 경과하면 하위 단계로 데이터의 이동이 일어나고, 환자의 이력 정보가 필요하면 제3의 저장 장치에서 상위 단계로 데이터 이동이 발생한다.

600병상 정도의 규모에서 4K CR을 기준으로 무손실압축 저장 용량은 1주일 기준으로 100GB 이상의

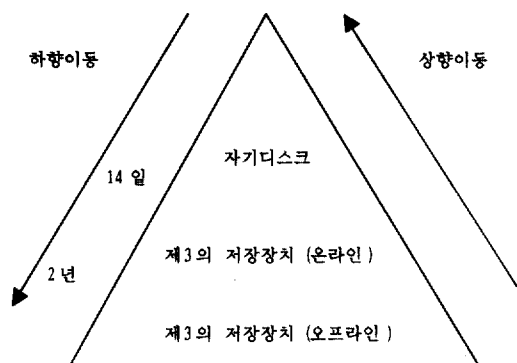



그림 1. 의료 영상 데이터의 계층 저장 구조

용량을 차지한다[12]. 일정한 보존 기간이 경과하면 단기 저장 장치에 저장되어 있는 영상을 압축하여 장기 저장 장치로 옮기는데, 장기 저장용 영상은 20대 1 또는 40대 1 정도로 압축해서 저장되고 있다. 기존의 의료 영상 데이터베이스에서는 일정한 보존 기간이 경과하면 보존 기간을 기준으로 의료 영상을 획일적으로 단기 저장 장치에서 장기 저장 장치로 옮기고 있다. 이러한 의료 영상 이동 방법은 다음과 같은 문제점이 있다.

첫째, 어떤 환자의 이전에 촬영한 영상이 최근에 촬영한 영상과 비교 진단에 쓰이는 경우에, 곧 비교 진단에 사용될 이전에 촬영한 영상마저 제3의 저장 장치에 옮겨져 있을 가능성이 많다. 비교 진단에 사용될 이러한 영상은 제3의 저장 장치에서 다시 자기 디스크로 옮겨야 하므로 지금의 기술에 의하면 최소한 10초 이상의 검색 시간이 소요된다.

둘째, 기존의 이동 방법은 개체 버전들 사이의 시간 연관성을 고려하지 않고 자기 디스크에서 제3의 저장 장치로 옮기므로 시간 연관성이 높은 개체 버전들이 서로 다른 여러 개의 CD에 흩어져서 저장될 가능성이 크다.

의료 영상 데이터베이스의 단기 저장 장치인 자기 디스크에서 일정한 보존 기간이 경과하면 제3의 저장 장치로 의료 영상을 옮기는 기존 이동 방법의 문제점을 보기를 들어서 살펴 보도록 한다. 그림 2에서 E_i는 환자 개체를 뜻하고 E_{ij}는 E_i의 j 번째 개체 버전을 나타내고 각 개체 버전은 이력 의료 영상을 포함하고 있다. 환자 레코드에서 ●으로 표시한 곳은 의료 영상을 촬영하여 데이터베이스에 기록한 시간을 나타낸다. E_{11}은 22일에 촬영한 영상을 포함하고 있는 E_1의 첫번째 개체 버전이고 E_{12}는 현재(30일) 촬영한 영상을 가지고 있는 E_1의 두 번째 개체 버전을 나타낸다.

보존 기간이 14일이라고 가정하면 개체 버전 E_{31}, E_{41}, E_{51}, E_{61}, E_{21}, E_{32}, E_{52}는 장기 저장 장치에 옮겨진 영상이다.개체 버전 E_{33}이 가지고 있는 의료 영상과 이전 영상을 가지고 있는 E_{32}의 영상을 비교 진단하는 경우에는 E_{32}의 영상을 제3의 저장 장치에서 자기 디스크로 이동시켜야 하므로 제3의 저장 장치가 CD 쥬크 박스인 경우에 10초 안팎의 시간이 소요된다. 또한, 개체 버전 E_{31}이 가지고 있는 의료 영상이 비교 진단에 필요하고 E_{31}의 영상이 다른 CD에 저장되어 있다면 추가의 10여 초 시간이 소요될 것이다. 만약에 개체 버전 E_{31}과 E_{32}가 가지고 있는 의료 영상을 제3의 저장 장치로 옮기지 않고 자기 디스크에 그대로 두었다면 수십 밀리 초 안에 검색이 완료될 것이다. E_{32}가 단기 저장 장치인 자기 디스크에 그대로 저장되어 있거나 E_{31}과 E_{32}가 같은 CD에 저장되어 있다면 10여 초 정도 검색 시간을 단축할 수 있다.

본 논문에서는 앞에서 지적한 기존 이동 방법의 두 가지 문제점을 해결하기 위한 이력 의료 영상 이동 방법을 제안한다. 먼저, 곧 비교 진단에 사용될 가능성이 높은 영상은 단기 저장 장치에 그대로 둠으로써 검색 속도를 향상시키고, 비교 진단에 쓰일 가능성이 낮은 영상은 제3의 저장 장치로 이동하여 공간의 효율성을 높이는 영상 데이터 관리기법을 제안한다. 또한, 개체 버전들 사이에 시간 연관성 등을 고려하지 않고 제3의 저장 장치로 이동하므로 시간 연관성과 무관하게 여러 세그먼트에 흩어져서 저장되는데, 이러한 경우에 시간 질의에 대한 성능이 저하된다. 본 논문에서는 시간 연관성이 높은 개체 버전들을 인접하게 저장하는 방법을 제안한다.

단위 : 일

	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	현재
E_1													● E_{11}			● E_{12}
E_2								● E_{21}				● E_{22}				● E_{23}
E_3	● E_{31}							● E_{32}								● E_{33}
E_4	● E_{41}															● E_{42}
E_5				● E_{51}				● E_{52}				● E_{53}				● E_{54}
E_6				● E_{61}												● E_{62}

그림 2. 의료 영상 기록 시간으로 나타낸 환자 개체버전의 보기

3. 이력 의료 영상 관리기법

본 절에서는 이력 의료 영상의 관리 기법으로써 EAT 이동 기법과 시간 연관성에 기반한 이력 영상의 저장 방법을 제안한다. 3.1절에서는 EAT 이동 기법에서의 경계값을 정의하고 각 영역에 저장되는 개체 버전을 정의한다. 그리고, 경계값을 구하고 개체 버전을 이동하는 알고리즘을 제시한다. 3.2절에서는 시간 연관성에 기반하여 개체 버전을 저장하기 위한 겹침의 정도와 간격의 정도를 정의하고, 두 값을 통합하여 개체 버전을 저장 장치에 배치하는 방법을 보인다.

3.1 EAT 이동 기법

환자의 기록에는 환자 정보와 의료 영상 데이터가 들어가는데 환자의 기록과 관련된 데이터는 주 키, 시간 데이터, 키가 아닌 데이터, 이차 데이터 등으로 나눌 수 있다. 본 논문에서는 주 키는 각 개체를 구분하는 식별자로 사용하고, 시간 데이터는 트랜잭션 시간을 사용한다. 트랜잭션 시간은 환자의 이력 데이터가 저장된 시간으로써 트랜잭션 시작 시간과 트랜잭션 종료 시간으로 구성된다. 키가 아닌 주 데이터에는 환자의 기본 정보를 나타내고 이차 데이터는 시간이 경과하면서 달라진 환자의 상태 데이터를 나타낸다. 본 논문에서 환자의 각 레코드는 비시간 데이터의 키값으로 인덱싱되어 있으며, 논리적인 저장 모델로 관계형 데이터 모델을 확장한 튜플 버저닝(tuple versioning)을 사용한다.

본 논문에서 사용하는 기호와 의미는 다음과 같다.

- T_s : 트랜잭션 시작 시간(transaction start time) [13].
- T_e : 트랜잭션 종료 시간(transaction end time) [13].
- E_{ij} : 개체 E_i의 j번째 버전
- $E_{ij}T_s$: 개체 버전 E_{ij}의 트랜잭션 시작 시간
- $E_{ij}T_e$: 개체 버전 E_{ij}의 트랜잭션 종료 시간
- now : 현재 시간

단기 저장 장치에서 장기 저장 장치로 의료 영상을 옮기기 위해서는 이동을 위한 경계값이 필요한데, 기존에는 현재 시간에서 2주일 전과 같이 일정한 시간을 경계값으로 사용하였다. 본 논문에서는 동적인 경계값을 생성하기 위해서 트랜잭션 시간을 이용한다. 먼저 기준값이 되는 트랜잭션 시간의 평균 간격을 구하면 다음과 같다.

$T = \{t_0, t_1, \cdots, t_x\}$는 시간값을 가지는 유한 집합이라 하고, $U = \{A_1, A_2, \cdots, A_n\}$는 트랜잭션 시간을 제외한 열의 집합이라고 하면 환자 테이블 스키마는 $R = \{U, T_s, T_e\}$로 정의한다. 단, $\{T_s, T_e\} \in T$ 이다. R에 들어있는 어떤 개체 버전 E_{ij}의 트랜잭션 시간 간격을 $E_{ij}T_l$ 이라고 하면 다음과 같이 정의한다.

$$E_{ij}T_l = E_{ij+1}T_s - E_{ij}T_s$$

개체 버전들의 트잰잭션 시간 간격의 집합을 I라고 하고 다음과 같이 정의된다.

$$I = \{E_{11}T_l,\ E_{12}T_l, \cdots,\ E_{ij}T_l, \cdots,\ E_{mn}T_l\}$$

트랜잭션 시간 간격의 집합 I에서 트랜잭션 시간 간격의 개수는 다음과 같이 정의한다.

$$n = |I| = |\{E_{11}T_l,\ E_{12}T_l, \cdots,\ E_{ij}T_l, \cdots,\ E_{mn}T_l\}|$$

개체 버전들의 전체 시간 범위 t_0에서 t_{now}까지 평균 트랜잭션 시간 간격(줄여서 "평균 간격(average interval)"이라고 한다)은 다음과 같다.

$$\sum_{i=1}^{m} \sum_{j=1}^{n} E_{ij}Tl / n = l$$

현재 시간에서 평균 간격 l을 뺀 시간을 p라고 하고 다음과 같이 정의한다.

$$p = now - l$$

기준 시점 p 에서 각 개체 버전들의 트랜잭션 시작 시간은 before, equal, 그리고 after 관계가 성립한다(그림 3).

연산자	의미	관계도
E_{ij} before p	$E_{ij}T_s < p$	E_{ij} — p
E_{ij} equal p	$E_{ij}T_s = p$	p / E_{ij}
E_{ij} after p	$E_{ij}T_s > p$	p — E_{ij}

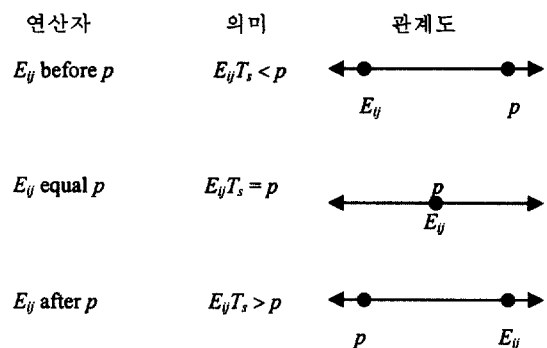



그림 3. 기준 시점과 각 개체 버전 트랜잭션 시간과의 관계

그림 3에서와 같이 기준 시점 p와 각 개체 버전이 가지고 있는 트랜잭션 시간과의 차이는 $|EijTsp| = d$로 정의한다. 장기 저장 장치의 저장 영역을 C, 자기 디스크의 저장 영역을 D라고 하면, C와 D에 저장되는 개체 버전을 구분하는 경계값 EAT(*Expanded Average Transaction time*)는 다음과 같이 정의한 다.

$$EAT = min\{E_{ij}T_s \mid (E_{ij}T_s < p) \wedge (d < l)\}$$

C에 저장되는 개체 버전의 집합을 $C(E)$, D에 저장되는 개체 버전의 집합을 $D(E)$ 라고 하고 각각 다음과 같이 정의한다.

$$C(E) = \{E_{ij} \mid E_{ij}T < EAT\}$$
$$D(E) = \{E_{ij} \mid E_{ij}T \geq EAT\}$$

그림 4는 저장 영역 D에 있는 개체 버전에 경계값 EAT 를 적용하여 저장 영역 C로 이동할 개체 버전을 결정하는 보기를 나타낸 것이다. 그림 4에서 l은 평균 간격이고 현재 시간인 *now*에서 평균 간격을 l을 뺀 시점이 p이다. 평균 간격의 특성 때문에 시점 p의 주변에 트랜잭션 시간이 많이 모이게 된다.

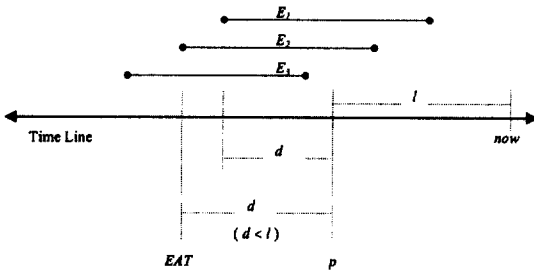



그림 4. 개체 버전 이동을 위하여 경계값을 적용한 보기

시점 p에서 평균 간격 l을 넘지 않으면서 최소의 트랜잭션 시작 시간을 가지는 개체 버전의 트랜잭션 시작 시간을 저장 영역의 경계값으로 한다. 그림 4에서 E_1, E_2, E_3가 모두 시점 p보다 트랜잭션 시작 시간이 작지만 E_3는 시점 p에서 평균 간격 l보다 크므로 경계값으로 채택되지 않는다. E_2는 E_1보다 트랜잭션 시작 시간이 작으면서 시점 p에서 평균 간격 l보다 작으므로 경계값이 된다.

E_2의 트랜잭션 시작 시간인 EAT가 저장 영역의 경계값이 되는데 EAT보다 트랜잭션 시작 시간이 작은 개체 버전은 C로 옮기고 EAT보다 트랜잭션 시작 시간이 크거나 같은 개체 버전은 D에 그대로 둔다. 경계값 EAT는 시간축 상에서 p의 주변에 모여있는 개체 버전 가운데 p보다 트랜잭션 시작 시간이 작은 개체 버전을 C로 옮기지 않고 D에 그대로 보관하는 효과를 가져온다. 또한, 경계값 EAT가 시점 p에서 평균 간격 l을 넘지 않으므로 비교 진단에 사용될 가능성이 낮은 이력 영상은 C로 옮기게 한다.

그림 5는 개체 버전이 저장되어 있는 세그먼트에서 평균 간격과 경계값을 구하는 알고리즘을 나타낸 것이며, 그림 6은 경계값을 기준으로 저장 영역 C로 옮길 개체 버전을 선택하고 이동하는 알고리즘을 나타낸 것이다.

3.2 시간 연관성에 기반한 저장 방법

3.1절의 EAT에 의한 이동 기법에서 살펴 본 경계

```
알고리즘1 GetBoundary( )
// 모든 개체버전 Eij가 저장되어 있는 세그먼트에 대해서
t1 ← t2 ← 0, interval ← 0, sum ← 0, now ← DateTime, t ← 0
while not end-of-file (segment) do
      EntityVersion1 ← read segment record
      EntityVersion2 ← read next record with same id
      t1 ← EijTs of EntityVersion1
      t2 ← Eij+1Ts of EntityVersion2
      interval ← t2 - t1
      sum ← sum + interval
      n ← n + 1
end while
if  sum = 0  then
      Message(Not found entity version)
else
      AvgLength ← sum / n
      AvgPoint ← now - AvgLength
end if
      TempBoundry ← AvgPoint + AvgLength
while not end-of-file (segment) do
      t ← 0
      EntityVersion ← read segment record
      t ← EijTs of EntityVersion
      if (t < AvgPoint and TempBoundry < t) then
      add (Buffer, t)
      end if
      boundary ← min(Buffer)
end while
return boundary
```

그림 5. 경계값을 구하는 알고리즘

알고리즘 2 Migration( )

while (not end-of-file(*segment*) **do**
  EntityVersion ← read segment record
  if($E_{ij}T_s$ of EntityVersion < *boundary*) **then**
   add(*Buffer*, *EntityVersion*)
   delete(*segment*, *EntityVersion*)
  end if
end while
If *Buffer* is null **then**
  Message(*Not found entity version for migration*)
  exit
else TimeInterrelationClustering($E_{ij}T_s$, $E_{ij}PK$, *Buffer*)
while(Buffer not empty) **do**
  move record to segment of tertiary storage
end while

그림 6. 저장영역C로 개체 버전을 옮기는 알고리즘

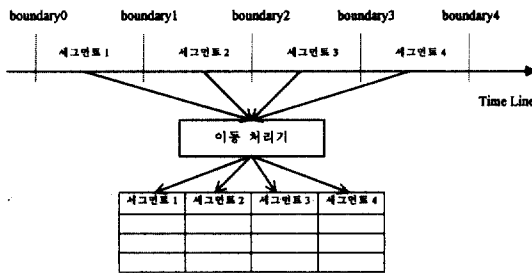



그림 7. 세그먼트를 나누는 경계값과 저장 개념도

값은 시간축 상에서 각 세그먼트를 구분하는 기준이 되며, 이동 처리기는 경계값에 의하여 이동 대상이 되는 개체 버전을 저장 영역 *C*로 옮기게 된다(그림 7). 경계값을 기준으로 개체 버전을 이동하면 저장 영역 *C*에 저장되는 세그먼트는 세그먼트 사이에 트랜잭션 시작 시간을 기준으로 하는 클러스터링이 이루어지게 되는데 아래에서 그 과정을 보이고, 이어서 시간 연관성에 의한 세그먼트 내의 저장 방법을 제안한다.

$C(E)$로 이동되는 각 세그먼트에 들어있는 개체 버전의 트랜잭션 시간 간격 가운데 가장 작은 트랜잭션 시작 시간을 찾는 함수를 *FindMinStart*( ), 가장 큰 트랜잭션 종료 시간을 찾는 함수를 *FindMaxEnd*( ) 라고 하면, $C(E)$로 이동되는 세그먼트에서 최소값과 최대값을 구할 수 있다. $C(E)$로 이동되는 어떤 세그먼트의 트랜잭션 시작 시간의 집합을 TSV_i, 트랜잭션 종료 시간의 집합을 TEV_i라 하고 다음과 같이 정의한다.

$$TSV_i = \{TransStartValueS_1, TransStartValueS_2, \cdots, TransStartValueS_i\}$$

$$TEV_i = \{TransEndValueS_1, TransEndValueS_2, \cdots, TransEndValueS_i\}$$

어떤 세그먼트의 트랜잭션 시작 시간의 최소값은 $FindMinStart(TSV_i)$이고, 트랜잭션 종료 시간의 최대값은 $FindMaxEnd(TEVi)$가 된다. $MinStart(TSV_i)$와 $FindMaxEnd(TEV_i)$를 각각 $MinTSV_i$, $MaxTEV_i$라고 하면, 어떤 세그먼트의 lifespan을 $L(S_i)$라고 하면, 다음과 같이 정의된다.

$$L(Si) = [MinTSV_i, MaxTEV_i]$$

$C(E)$의 i 번째 세그먼트의 $MinTSV_i$와 $i+1$ 번째 세그먼트의 $MinTSV_{i+1}$는 다음 관계가 성립한다.

$$MinTSV_i < MinTSV_{i+1}$$

$C(E)$의 i 번째 세그먼트의 lifespan은 $L(S_i)$이고, $i+1$번째 세그먼트의 lifespan은 $L(S_{i+1})$가 된다. $L(S_i)$의 시작 시간은 $MinTSV_i$이고 $L(S_{i+1})$의 시작 시간은 $MinTSV_{i+1}$이며 각각의 세그먼트에 들어있는 개체버전은 세그먼트 사이에 중복되지 않는다. 따라서 $C(E)$로 옮겨진 세그먼트들은 $MinTSV_i$를 기준으로 세그먼트 사이에 시간순 클러스터링이 된다. 또한, $MinTSV_{i+1}$과 $MinTSV_i$의 차이는 다음과 같다.

$$MinTSV_{i+1} - MinTSV_i(d+l)$$

한편, $C(E)$에 저장되어 있는 모든 세그먼트의 lifespan은 $L(L(S_1) \cup L(S_2) \cup \cdots \cup L(S_i))$가 된다.

그림 8에서 세그먼트 1, 세그먼트 2, 세그먼트 3, 세그먼트 4 각각의 $MinTSV_1$, $MinTSV_2$, $MinTSV_3$, $MinTSV_4$는 각각 20, 40, 60, 80임을 나타낸 것인데, 네 개의 세그먼트 사이에 트랜잭션 시작 시간 순으로 세그먼트 간 클러스터링이 됨을 알 수 있다.

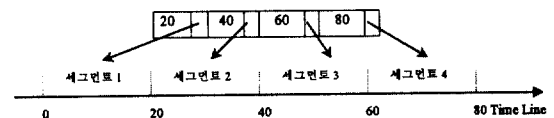



그림 8. 세그먼트 사이의 트랜잭션 시작 시간순 클러스터링 개념도

의료 데이터베이스에서 환자의 이력 정보와 관련된 시간 질의(temporal query)는 트랜잭션 시점 질의와 시간 간격 질의가 있고 식별자 중심 질의가 있다.

● 트랜잭션 시점 질의

- 입력되는 질의가 트랜잭션 시점(point)이 되며 출력되는 결과는 시점을 포함하는 트랜잭션 시간 간격이 된다(그림 9). 어떤 시점에 촬영한 모든 환자의 상태를 검색하는 질의를 말한다.

● 트랜잭션 시간 간격 질의

- 입력되는 질의에서 트랜잭션의 시간이 간격으로 주어지며, 출력되는 결과도 시간 간격으로 표현된다. 입력 시간 간격과 출력 시간 간격 사이에는 Containment, Inclusion, Intersection 가운데 어느 하나가 된다(그림 9). 어떤 시간 간격에 촬영한 모든 환자의 상태를 검색하는 질의를 말한다.

● 식별자 중심 질의

- 특정 식별자를 가지는 개체 버전을 검색하는 질의로써, 특정 환자의 이력 정보를 검색하는 질의를 말한다.

질의분류	트랜잭션 시점 질의	트랜잭션 시간 간격 질의		
질의형태	Point	Containment	Inclusion	Intersection
입 력	T	T_s T_e	T_s T_e	T_s T_e
가능한 출력				

그림 9. 트랜잭션 시점 및 시간 간격 질의의 형태

본 논문에서는 트랜잭션 시점 질의와 트랜잭션 시간 간격 질의를 환자의 이력 정보에 대한 질의로 선정하고 이들 질의에 대해서 시간 연관성에 의한 세그먼트 내의 개체 버전 저장 방법을 제안한다.

세그먼트 안에서 어떤 두 개체 버전이 가지고 있는 트랜잭션 시간은 겹치거나 간격을 두고 있는데, 겹침의 정도와 간격의 정도를 정의하면 다음과 같다.

겹침의 정도

$$DOV(E_1, E_2) = \begin{cases} min(E_1T_e, E_2T_e) - max(E_1T_s, E_2T_s) & \text{단, } E_1T_e \geq E_2T_s \wedge E_2T_e \geq E_1T_S \\ 0 & \text{그 외} \end{cases}$$

간격의 정도

$$DOD(E_1, E_2) = \begin{cases} E_2T_s - E_1T_e & \text{단, } E_1T_e < E_2T_s \\ 0 & \text{그 외} \end{cases}$$

겹침의 정도와 간격의 정도는 두 개체 버전이 시간적으로 얼마나 연관성이 있는가를 계량화한 것이다. 트랜잭션 시점 질의와 트랜잭션 간격 질의에 대한 겹칩의 정도와 간격의 정도와의 연관성으로부터 다음과 같은 두 가지 사실을 알 수 있다.

첫째, 트잰잭션 시점 질의가 선택될 확률은 $DOV(E_1, E_2)$에 비례한다.

E_1, E_2, E_3을 개체 버전이라 하고 $DOV(E_1, E_2) = x$, $DOV(E_1, E_3) = y$, $x>y$ 라고 가정한다. E_1, E_2, E_3의 lifespan을 포함하는 전체 lifespan을 z라고 하고, E_1, E_2가 동시에 선택될 확률을 p_1이라고 하면,

$$p_1 = \frac{x}{z}$$

E_1, E_3이 동시에 선택될 확률을 p_2라고 하고, 같은 방법으로 p_2를 구하면 다음과 같다.

$$p_2 = \frac{y}{z}$$

가정에서 $x>y$ 이므로, $p_1>p_2$가 되어서 트잰잭션 시점 질의가 선택될 확률은 $DOV(E_1, E_2)$에 비례함을 알 수 있다. 보기를 들면, 다음과 같다.

전체 lifespan을 [0, 100]이라 하고, 세 개의 개체 버전 E_1, E_2, E_3의 트랜잭션 시간을 [30, 50], [20, 40], [45, 55]라고 하면, $DOV(E_1, E_2) = 10$, $DOV(E_1, E_3) = 5$ 가 된다. $p_1 = 10/100$, $p_2 = 5/100$이므로 $DOV(E_1, E_2)$가 $DOV(E_1, E_3)$보다 크면, E_1, E_2가 동시에 선택될 확률이 E_1, E_3가 같이 선택될 확률보다 높음을 알 수 있다.

둘째, 트랜잭션 시간 간격 질의가 선택될 확률은 $DOV(E_1, E_2)$에 비례하고, $DOD(E_1, E_2)$에 반비례한다.

E_1, E_2, E_3을 개체 버전이라 하고 $DOV(E_1, E_2) = x$, $DOV(E_1, E_3) = y$, $x>y$라고 가정한다. 전체 lifespan을 w라 하고, E_1, E_2가 동시에 선택될 확률을 p_1이라고 하면,

$$p_1 = \frac{x}{w}$$

E_1, E_3가 동시에 선택될 확률을 p_2라고 하고, 같은 방법으로 p_2를 구하면 다음과 같다.

$$p_2 = \frac{y}{w}$$

가정에서 $x>y$ 이고, $p_1>p_2$가 되므로 트랙잭션 시간 간격 질의가 선택될 확률은 $DOV(E_1, E_2)$에 비례

한다.

그리고, E_1, E_2, E_3을 개체 버전이라 하고 $DOD(E_1, E_2) = a$, $DOD(E_1, E_3) = b$, $a>b$라고 가정하자. 전체 lifespan을 w라 하고, E_1, E_2가 동시에 선택되지 않을 확률을 p_1이라고 하면, 다음과 같다.

$$p_1 = \frac{a}{w}$$

따라서, E_1, E_2가 동시에 선택될 확률은 $1-p_1$이 된다.

E_1, E_3가 동시에 선택되지 않을 확률을 p_2라고 하고, 같은 방법으로 p_2를 구하면,

$$p_2 = \frac{b}{w}$$

따라서, E_1, E_3이 동시에 선택될 확률은 $1-p_2$가 된다.

가정에서 $a>b$ 이고, $1-p_2>1\ p_1$이 되므로 트랙잭션 시간 간격 질의가 선택될 확률은 $DOD(E_1, E_2)$에 반비례함을 알 수 있다. 보기를 들면 다음과 같다.

트랜잭션 시간 간격 질의가 선택될 확률은 $DOV(E_1, E_2)$에 비례하는 보기는 앞에서 살펴 본 보기 1과 같고, $DOD(E_1, E_2)$에 반비례하는 보기를 들도록 한다.전체 lifespan을 [0, 100]이라 하고, 세 개의 개체 버전 E_1, E_2, E_3의 트랜잭션 시간을 [10, 30], [60, 70], [40, 50]라고 하면, $DOV(E_1, E_2) = 30$, $DOV(E_1, E_3) = 10$ 이 된다. E_1, E_2가 동시에 선택되지 않을 확률 $p_1 = 30/100$ 이고 E_1, E_3가 동시에 선택되지 않을 확률 $p_2 = 10/100$이므로 E_1, E_3가 동시에 선택될 확률 $1-p_2$가 E_1, E_2가 동시에 선택될 확률 $1-p_1$보다 큼을 알 수 있다.

개체 버전 사이에 겹침의 정도와 간격의 정도를 계량화하였고 트잭잭션 시점 질의가 선택될 확률은 겹침의 정도에 비례함을 보였으며, 트랜잭션 시간 간격 질의가 선택될 확률은 겹침의 정도에 비례하고, 간격의 정도에 반비례함을 보였다.

그림 10은 앞에서 살펴 본 겹침의 정도와 간격의 정도에 기반하여 세그먼트 안에서 트랜잭션 시간 순으로 개체 버전이 배치됨을 보기로 나타낸 것이다. 또한, 이 그림에서는 세그먼트 안에서 개체 버전의 배치 뿐만 아니라 본 논문에서 제안한 이동 방법에 의하여 세그먼트 사이에 트랜잭션 시간 순으로 세그먼트가 배치되는 것을 알 수 있다.

겹침의 정도와 간격의 정도를 세그먼트에 저장되는 개체 버전들에게 적용하기 위해서는 통합된 하나의 시간 연관성에 관한 식이 필요한데, 다음은 그 식을 나타낸 것이다.

$$I(E_1, E_2) = \alpha \times DOV(E_1, E_2) + \beta \times DOD(E_1, E_2)$$

위 식에서 α와 β는 가중치로써 시간지원 질의의 개수 가운데 트랜잭션 시점 질의 개수와 트랜잭션 시간 간격 질의 개수의 비율을 나타낸 것이다. 시간지원 질의 수 가운데 트랜잭션 시점 질의 개수를 p, 트랜잭션 시간 간격 질의 개수를 q라 하면, $\alpha = p / (p+q)$이고 $\beta = q / (p+q)$가 된다.

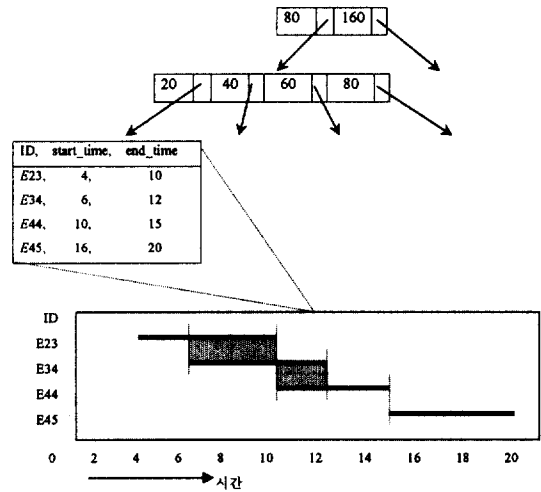



그림 10. 세그먼트 사이 및 세그먼트 안의 트랜잭션 시간 순 배치 보기

4. 성능 평가

본 절에서는 기존의 이동 방법과 본 논문에서 제안한 이동 방법 및 시간 연관성 기반한 저장 방법 사이에 실험을 통하여 성능을 비교하고 분석한다. 4.1절에서는 실험을 위한 환경에 대해서 살펴보고, 4.2절에서는 실험의 결과를 평가한다.

4.1 실험 환경

병원의 의료 영상은 효율적인 검색과 저장을 위해서 단기 저장 장치와 장기 저장 장치로 나누어서 저장한다. 병원에 따라 따르나 2주일 정도의 의료 영상을 단기 저장 장치인 자기 디크에 저장하고 일정한 보존 기간이 경과하면 단기 저장 장치에서 장기 저장 장치인 CD주크 박스나 DLT 등으로 의료 영상을 옮겨서 저장한다.

일정한 기간마다 의료 영상을 옮기는 기존 이동 방법과 본 논문에서 제안한 EAT에 의한 이동 방법 및 시간 연관성에 기반한 저장 방법 사이에 성능을 평가하기 위해서, 본 논문에서는 각각의 방법에 따라 제3의 저장 장치에서 디스크로 데이터가 옮겨졌다고 가정하고 실험한다. 기존의 이동 방법에서는 환자의 식별자로 인덱싱이 되었다고 가정한다. 본 논문에서 제안한 방법은 EAT에 의한 이동 방법 및 시간 연관성에 기반한 저장 방법으로서 앞에서 살펴 본대로 겹침의 정도과 간격의 정도를 통합한 시간 연관성을 이용하여 저장되었다고 가정한다. 두 방법 사이에 성능을 평가하기 위해서 시점 질의와 시간 간격 질의에 대해서 클러스터의 참조 회수를 비교하도록 한다.

시간 데이터베이스의 전체 시간 범위는 3,650으로 하였으며 단위는 1일로 하였다. 트랜잭션 시간 길이는 1에서 90사이에서 단위가 1인 크기로 균등분포로 생성하였다. 500병상 규모의 병원에서 하루에 약 400건 정도의 촬영이 있으므로 3,650일 동안에 1,460,000개의 데이터가 생성될 것으로 추정하여 실험에 사용한 데이터의 개수는 1,460,000개로 하였다.

의료 영상은 무압축시 8MB정도가 되는데 장기 저장 장치에 저장되는 크기는 40 : 1로 압축된 것으로 가정하여 클러스터의 수는 456개로 설정하였다.실험에서 질의는 시점 질의와 시간 간격 질의를 각각 10%, 20%, 30%, 40%, 그리고 50%로 생성하여 각각 실험하였고, 또한, 시점 질의와 시간 간격 질의를 같은 비율로 생성하여 시간 질의를 10%, 20%, 30%, 40%, 그리고 50%로 변화하면서 실험하였다.

4.2 실험 결과

그림 11과 그림 12는 각각 시점 질의와 시간 간격 질의를 10%에서 50%까지 변화시키면서 클러스터의 참조 회수를 나타낸 것이다. 각 그림의 범례에서 T-Cluster는 EAT 이동 방법 및 시간 연관성에 기반한 저장 방법을 나타내고 K-index는 환자의 식별자로 인덱싱된 기존의 이동 방법에 의해서 저장된 것을 나타낸 것이다.

그림 11과 그림 12에서 시간 질의의 비율이 증가하면서 기존의 방법이 제안한 방법보다 클러스터의 참조 회수가 급격히 늘어나는 것을 알 수 있다. 이것은 기존의 방법이 개체 버전 사이에 시간 연관성과 무관하게 저장하였으므로 시간 연관성이 있는 개체 버전들이 여러 세그먼트에 흩어져서 저장되었기 때문이다. 그림 11보다 그림 12에서 전체적으로 클러스터의 참조 회수가 많은 것은 시점 질의보다 시간 간격의 질의가 두 개 이상의 세그먼트를 검색할 가능성이 높기 때문이다.

그림 13은 시점 질의와 시간 간격 질의를 같은 비율로 생성하고 시간 질의를 10%에서 50%까지 변화시키면서 클러스터의 참조 회수를 비교한 것이다. 시간 질의가 증가할수록 기존의 이동 방법의 클러스터 참조 회수가 제안한 방법보다 현저히 증가함을 알 수 있다. 시점 질의와 시간 간격 질의가 같은 비율로 수행되므로 클러스터 참조 회수의 전체적인 흐름을 보면, 그림 11의 시점 질의보다는 클러스터 참조 회수가 많고 그림 12의 시간 간격 질의보다는 클러스터의 참조 회수가 적게 나타났다.

5. 결 론

병원의 의료 영상 데이터는 일정한 보존 기간이 경과하면 자기 디스크에서 제 3의 저장 장치로 영상을 옮기게 되는데, 기존의 획일적인 이동 방법은 곧

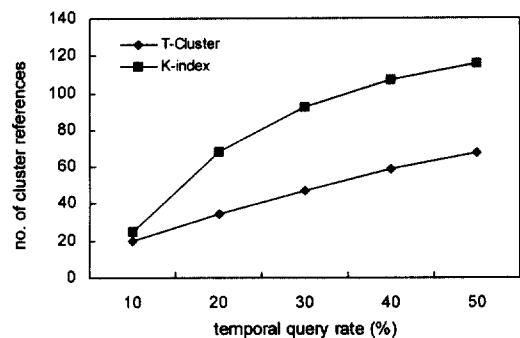



그림 11. 시점 질의 변화에 대한 클러스터 참조 회수

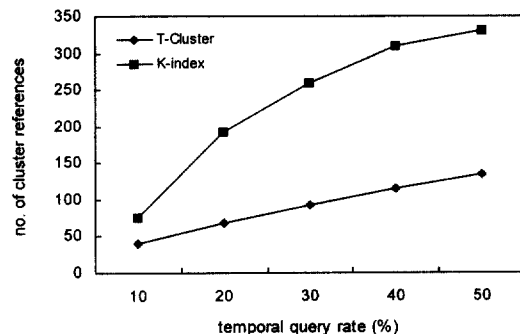



그림 12. 시간 간격 질의 변화에 대한 클러스터 참조 회수

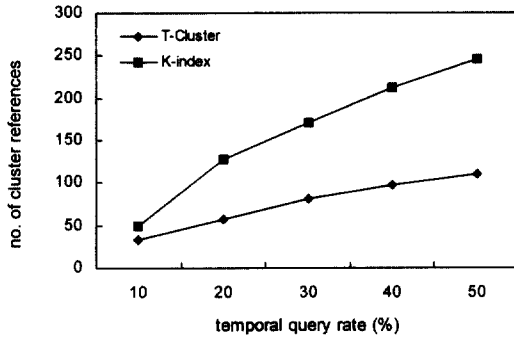



그림 13. 시점 및 시간 간격 질의 변화에 대한 클러스터 참조 회수

검색할 가능성이 높은 이전의 영상마저 옮김으로써 검색 속도를 저하시킨다. 또한, 시간 연관성이 높은 개체 버전들이 여러 저장 영역에 흩어져서 저장될 가능성이 크다.

본 논문에서는 일정한 보존 기간이 경과하면 획일적으로 이동하는 방법에서 생기는 두 가지 문제점을 해결하기 위해서 의료 영상 데이터의 관리 기법을 제안하였다. 의료 영상 데이터의 관리 기법으로써 EAT 데이터 이동 기법과 시간 연관성 기반 데이터 저장 방법을 제안하였다. EAT 데이터 이동 기법에서는 자기 디스크와 제 3의 저장 장치에 저장되는 개체 버전을 구분하는 경계값과 각 영역에 저장되는 개체 버전을 정의하였다. 그리고, 경계값을구하고 데이터를 이동하는 알고리즘을 제시하였다. 시간 연관성에 기반한 데이터 저장 방법에서는 임의의 두 개체 버전에 대한 시간 연관성을 계량화하기 위하여 겹침의 정도와 간격의 정도를 정의하였고, 두 값을 통합하여 개체 버전을 저장 장치에 배치하는 방법을 보였다.

본 논문에서 제안한EAT 이동 기법 및 시간 연관성 기반 데이터 저장 방법의 우수함을 입증하기 위하여 기존의 방법과 제안한 방법의 성능을 비교하였다. 시간 질의의 비율을 변화시키면서 클러스터의 참조 회수를 비교한 결과, 기존의 방법보다 제안한 방법의 클러스터 참조 회수가 적게 나타남으로써 제안한 방법의 성능이 우수함을 보였다.

참 고 문 헌

[1] S. Hludoy and G.Noelle, PACS for Tele radiology, Proc. of 12th IEEE Symposium on Computer-Based Medical System, Stanford, Connecticut, pp.18-20, June, 1999.

[2] 탁계례, 김우생, 이상범, 의료영상 저장 전송시스템(PACS)의 발전 및 연구 동향, 정보과학회지, 제16권, 제12호, pp.6-8, 1998.

[3] 송군식, 신명진, 이수현, 이주희, 오용호, 서울중앙병원의 PACS 구축 및 운영 현황, 대한 PACS 학회지, 제4권, 제1호, pp.51-59, 1998.

[4] 차순주, 이동호, 김용훈, 허감, 신설 병원의 완전 PACS 도입 후 1년, 대한PACS학회 2001춘계학술대회, 2001.

[5] 강호영, 신소영, 중소형 병원에서의 Full PACS 설치 및 운영의 경험, 대한PACS학회 2000추계학술대회, 2000.

[6] 차순주 외 7인, PACS에 이용되는 압축영상의 화질 평가: 4K 고해상도 CR영상의 장기저장을 위한 적정 압축율, 대한PACS학회지, 제7권, 제1호, pp.9-12, 2001.

[7] 대한방사선협회, http://www.krta.or.kr, 최신방사선기술, 2001.

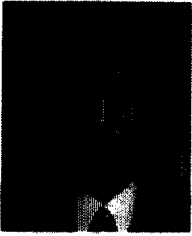
[8] 권대철, 변호영, 엄준희, 박범, Web기반의 원격 병리 진단 시스템과 PACS의 비교, 대한PACS 학회지, 제7권, 제1호, pp.25-33, 2001.

[9] 송군식, 이수현, 어기승, 윤경한, PACS운영중 발생하는 문제와 해결방안, 대한PACS학회지, 제6권, 제1호, pp.19-22.

[10] 차순주, 김용훈, 김수영, 황윤준, 김형석, 이동호, 허감, 고해상도 CR영상의 압축율에 따른 화질 비교, 대한PACS학회지, 제5권, 제1호, pp.10-18, 2000.

[11] 대한방사선협회, http://www.krta.or.kr.

[12] 하두희 외 9인, 디지털 영상 환경이 미비했던 기존 병원에서의 Full PACS도입: 분단 차병원의 사례, 대한PACS학회 2000추계학술대회, 2000.

[13] C.S. Jensen, R. Elmasri, S.K. Gadia, P. Hayes, and S. Jajodia(eds), A Glossary of Temporal Database Concepts, ACM SIGMOD Record, vol.6, no. 1, pp.52-64, March 1994.

윤 홍 원

1986년 부산대학교 계산통계학과 졸업(학사)
1990년 한국외국어대학교 경영정보대학원 전자계산학과(이학사)
1998년 부산대학교 대학원 전자계산학과(이학박사)
1996년~현재 신라대학교(구.부산여자대학교) 컴퓨터정보공학부 조교수
관심분야 : 의료정보시스템, 데이터베이스시스템, 시간데이터베이스, 멀티미디어시스템, XML
E-mail : hwyun@silla.ac.kr